



SIMULASI *ROUTING* PROTOKOL BERBASIS *DISTANCE VECTOR* MENGGUNAKAN GNS3 VERSI 0.8.

Ade Nurhayati & Adam Pangestu

Akademi Teknik Telekomunikasi Sandhy Putra Jakarta

Jalan Daan Mogot KM 11 Jakarta Barat 11710

E-mail: ade_nurhayati13@yahoo.com

ABSTRACT

Measurement of routing protocol performance in a network can be done with the help of simulation. The writers are trying to simulate package delivery data over Virtual Private Network (VPN) with the use of Graphical Network Simulator version 0.8.6 (GNS3). The topology used in the delivery of data packets is a ring topology which consists of 6 routers based on distance vector routing protocol. The results of simulation that the writers get are the throughput of the wireless and wireline on the server and client were averaging 1Mbit/s. The packet loss values generated for both wireless and wireline were 0 %. The measurement delay of the wireless and wireline for server and client side were averaging 7 ms.

Keywords: *RIP (Routing Information Protocol), VPN (Virtual Private Network), GNS3 (Graphical Network Simulator), wireshark, throughput, delay, packet loss.*

ABSTRAK

Pengukuran performansi routing protocol dalam suatu network dapat dilakukan dengan bantuan simulasi. Penulis mencoba mensimulasikan pengiriman paket data melalui Virtual Private Network (VPN) menggunakan simulator Graphical Network Simulator versi 0.8.6 (GNS3). Topologi yang penulis gunakan dalam pengiriman data paket menggunakan topologi ring yang terdiri dari 6 router, dengan salah satu routing protokol yang berbasis distance vector. Hasil simulasi yang penulis dapatkan nilai throughput dari wireless maupun wireline pada sisi server dan client rata-rata 1Mbit/sec. Sedangkan untuk nilai packet loss dihasilkan 0% baik itu wireless dan wireline dari posisi server ataupun client. Pengukuran pada delay wireless maupun wireline pada sisi server dan client rata-rata 7ms.

Kata kunci: *RIP (Routing Information Protocol), VPN (Virtual Private Network), GNS3 (Graphical Network Simulator), wireshark, throughput, delay, packet loss*

1. PENDAHULUAN

Perkembangan telekomunikasi dan informasi dunia sangat pesat, khususnya teknologi jaringan komputer dan internet telah mengalami perkembangan yang maju, sehingga mampu menyambungkan berbagai *host* komputer di dunia ini melalui jaringan yang ada untuk saling bertukar informasi satu sama lain. Alat transmisi dan media komunikasi itu pun semakin canggih dalam segi implementasinya dan seiring pesatnya perkembangan telekomunikasi dan informasi muncul jaringan berbasis *Internet Protocol* (IP). Untuk melayani pengguna khususnya dalam layanan data berbasis IP, banyak perangkat yang dioperasikan salah satunya pada bahasan ini adalah *router*. *Router* ini digunakan sebagai media pengiriman data dari sumber ke tujuan atau sebagai penghubung antar *network*.

Dalam implementasi pengiriman paket data dari sumber ke tujuan, *router* menggunakan *routing protocol*, salah satunya *Routing Information Protocol* (RIP). Pesatnya perkembangan telekomunikasi dan informasi mendorong kebutuhan user akan informasi pun semakin meningkat sehingga menyebabkan pembuat aplikasi atau para vendor serta pengusaha telekomunikasi semakin gencar mengembangkan produk unggulan untuk memenangkan persaingan. Menariknya perkembangan teknologi dan informasi membuat banyak pelajar muda ingin terjun ke dunia telekomunikasi dan mempelajarinya. Namun, untuk mengetahui pengoperasian *router* tersebut tidak mudah terwujud karena untuk mempelajari sistem komunikasi atau mendapatkan sertifikat itu mahal dan hanya penyelenggara-penyelenggara ICT (*Information Communication Telecommunication*) besar yang memiliki perangkat komunikasi untuk dipelajari.

Oleh karena itu penulis mensimulasikan cara pengoperasian *router* untuk pengiriman layanan data protokol *RIP* menggunakan GNS3 versi 0.8.6, agar para *engineer* yang akan terjun ke dalam dunia telekomunikasi dapat mempelajari dengan mudah dan murah tanpa harus memiliki *router* terlebih dahulu, serta para pekerja muda dapat menerapkannya dalam dunia kerja.



2. KAJIAN PUSTAKA

2.1. Protokol RIP (*Routing Information Protocol*)

Routing protokol yang menggunakan algoritma *distance vector*, yaitu algoritma *Bellman Ford* pertama kali dikenalkan pada tahun 1969 dan merupakan algoritma *routing* yang pertama pada ARPANET. Versi awal dari *routing protocol* ini dibuat oleh Xerox Parc's PARC *Universal Packet Internetworking* dengan nama *Gateway Internet Protocol*. Kemudian diganti nama menjadi *Router Information Protocol* (RIP) yang merupakan bagian *Xerox network Service* [1].

RIP merupakan *IP routing dynamic* untuk *distance vector* protokol di mana data disampaikan antar *network* berdasarkan jumlah *hop*. Jumlah *hop* router yang mampu dilalui RIP sebanyak 15 sebagai *routing metric* sedangkan *broadcast traffic* data di *update* setiap 30 detik untuk semua RIP *router* untuk menjaga integritas. Untuk menghindari *loop routing* digunakan teknik *split horizon with poison reverse*. RIP merupakan *routing protocol* yang paling mudah untuk dikonfigurasi [2].

Kelebihan RIP adalah menggunakan metode *Triggered Update*. RIP memiliki *timer* untuk mengetahui kapan *router* harus kembali memberikan *informasi routing*. Jika terjadi perubahan pada jaringan, sementara *timer* belum habis, *router* tetap harus mengirimkan *informasi routing* karena dipicu oleh perubahan tersebut (*triggered update*). Mengatur *routing* menggunakan RIP tidak rumit dan memberikan hasil yang cukup dapat diterima terlebih jika jarang terjadi kegagalan *link* jaringan.

Kelebihan RIP di antaranya:

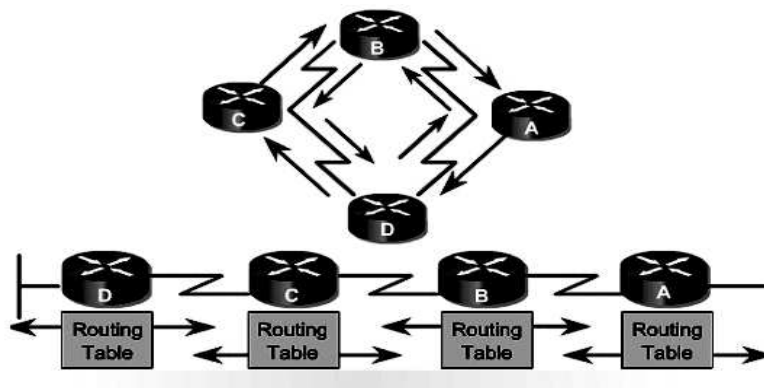
1. Hanya mengenalkan alamat yang terhubung langsung dengan *routernya*.
2. Tidak perlu mengetahui semua alamat jaringan yang ada.
3. Bila terjadi penambahan suatu jaringan baru, tidak perlu semua *router* mengkonfigurasi, melainkan hanya *router-router* yang berkaitan.

Berikut ini adalah cara kerja *Routing* RIP:

1. *Host* merespon pada alamat *broadcast* jika ada *routing update* dari *gateway*.
 2. *Host* akan memeriksa terlebih dahulu *routing table* lokal jika menerima *routing update*.
 3. Jika rute belum ada, informasi segera dimasukkan ke *routing table*.
-

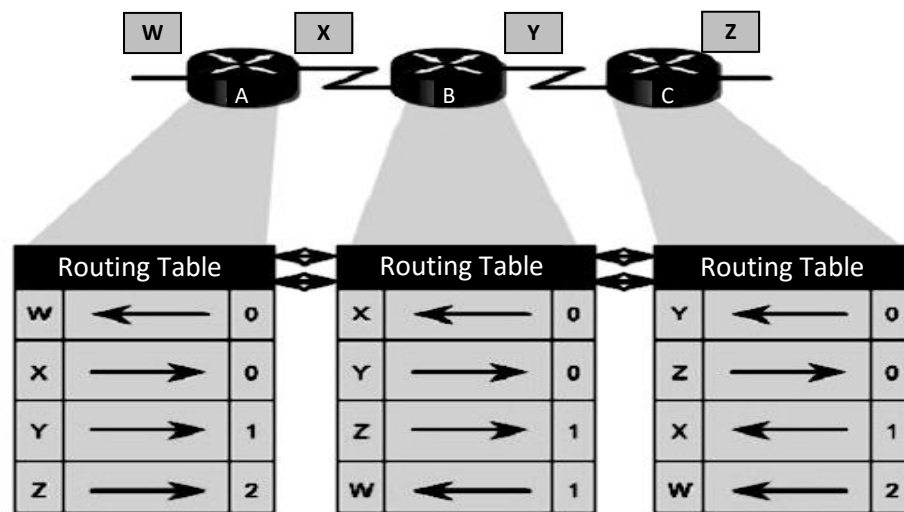
4. Jika rute sudah ada, *metric* yang terkecil (*hop*) akan diambil sebagai acuan.

Gateway RIP akan mengirimkan *routing update* pada alamat *broadcast* untuk setiap *network* yang terhubung. *RIP* mengirim pesan *routing update* secara berkala, dan ketika ada perubahan topologi jaringan, *routing* dapat menemukan rute baru, karena *router* menginformasikan *routing update* pada jaringan lain dari perubahannya. Berikut adalah contoh implementasi *routing RIP* seperti diperlihatkan pada Gambar 1.



Gambar 1. Contoh implementasi routing RIP [3]

Router B menerima informasi dari Router A. Router B menambahkan nilai distance vector bernilai 1 untuk jumlah hop, dan router B melewati table routing baru ke router C. Proses ini akan terus berlangsung untuk semua router tetangga. Algoritma distance vector melakukan update tiap 30 detik dan mengakumulasi jarak hop jaringan sehingga dapat digunakan untuk memperbaiki database informasi mengenai topologi jaringan, akan tetapi tidak mengizinkan router lain mengetahui secara pasti topologi internetwork, atau data pengguna di dalamnya. Router baru yang menggunakan distance vector pertama kali mengidentifikasi router tetangga. Cara router distance vector untuk menentukan jalur terbaik ialah berdasarkan informasi yang diterima dari tetangganya. Sebagai contoh, router A mencatat jaringan lain dari informasi yang diterima oleh router B. Masing-masing router menambahkan tabel routingnya dengan mencatat data router di sekelilingnya. Cara ini mempunyai fungsi untuk melihat sejauh mana jaringan yang akan dituju, seperti yang dijelaskan oleh Gambar 2 dibawah ini.

Gambar 2. Jaringan *distance vector* yang konvergen [3]

Dari penjelasan di atas dapat disimpulkan karakteristik RIP sebagai berikut:

1. Merupakan *distance vector routing protocol*
2. *Hop count* sebagai *metric* untuk memilih rute
3. Maksimum *hop count* 15, hop ke 16 dianggap *unreachable*
4. Secara *default routing update* 30 detik sekali.

2.2. GNS3

GNS3 (*Graphic Network Simulator*) adalah *software* simulasi jaringan komputer berbasis GUI yang mirip dengan *Cisco Packet Tracer*. Namun GNS3 memungkinkan simulasi jaringan yang kompleks, karena menggunakan *operating system* asli dari perangkat jaringan seperti Cisco dan Juniper, sehingga kita berada kondisi lebih nyata dalam mengkonfigurasi *router* langsung dari pada *Cisco Packet Tracer*. GNS3 adalah alat pelengkap yang sangat baik untuk laboratorium nyata bagi *network engineer*, administrator dan orang-orang yang ingin belajar untuk sertifikasi seperti Cisco CCNA, CCNP, CCIP dan CCIE serta Juniper JNCIA, JNCIS dan JNCIE [4][5].

Komponen GNS3 terdiri dari:

1. *Dynamips*, terkenal sebagai *Cisco IOS emulator*.

2. *VirtualBox*, menjalankan *desktop* dan sistem operasi *server* yang diketahui sebagai Juniper JUNOS.
3. *Qemu*, generik mesin *open source emulator*, berjalan sebagai Cisco ASA, PIX dan IPS.

2.3. Wireshark

Wireshark merupakan salah satu *tools* atau aplikasi “*Network Analyzer*” atau Penganalisis Jaringan. Cara kerja penganalisis kinerja jaringan itu dapat melingkupi berbagai hal mulai dari proses menangkap paket-paket data atau informasi yang lalu-lalang dalam jaringan sampai pada *sniffing* (memperoleh informasi penting seperti *password email*, dan lain-lain). *Wireshark* sendiri merupakan *free tools* untuk *Network Analyzer* yang ada saat ini. Tampilan *wireshark* ini sendiri terbilang sangat bersahabat dengan *user* karena menggunakan tampilan grafis atau GUI (*Graphical User Interface*) [6].

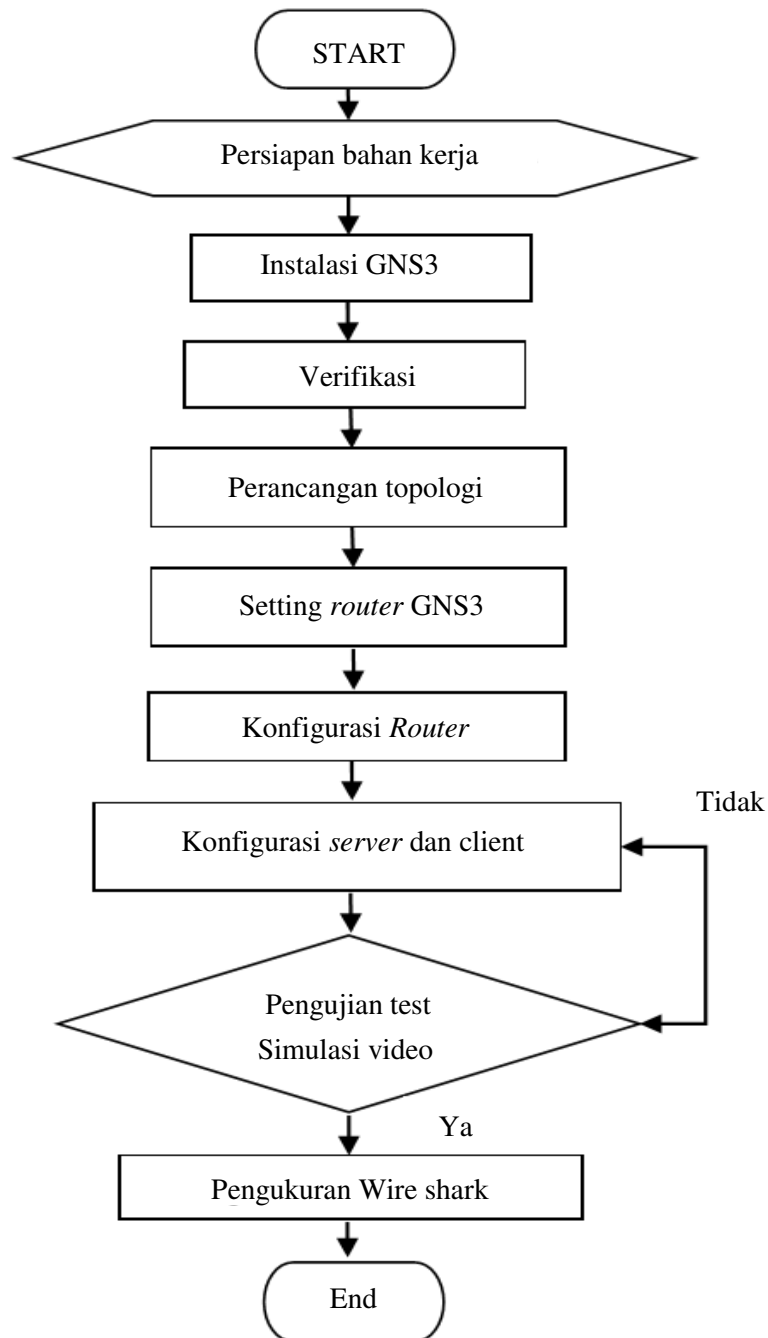
2.3.1. Kegunaan Wireshark

- a. Menganalisis jaringan.
- b. Menangkap paket data atau informasi yang berkeliaran dalam jaringan yang terlihat.
- c. Menganalisis informasi yang didapat dengan melakukan *sniffing*, dengan begitu dapat diperoleh informasi penting seperti *password*, dan lain-lain.
- d. Membaca data secara langsung dari *Ethernet*, *Token-Ring*, *FDDI*, *serial (PPP dan SLIP)*, *802.11 wireless LAN*, dan koneksi ATM.
- e. Dapat mengetahui IP seseorang.
- f. Menganalisis transmisi paket data dalam jaringan, proses koneksi, dan transmisi data antar komputer, dan lain-lain.

3. PROSES SIMULASI JARINGAN

3.1. Simulasi Layanan Data dengan Protokol RIP

Proses simulasi yang dilakukan adalah sebagai berikut:



Gambar 3. *Flowchart* proses simulasi

3.2. Langkah-langkah Simulasi Protokol RIP

Berikut adalah langkah-langkah dalam membangun simulasi protokol RIP untuk layanan data:

1. Penentuan dan perhitungan IP pada *router* GNS3

172.16.10.0/30

Binary	128	64	32	16	8	4	2	1
	128	192	224	240	248	252	254	255
/30 =	11111111	11111111	11111111	11111111	11111111	11111111	00	

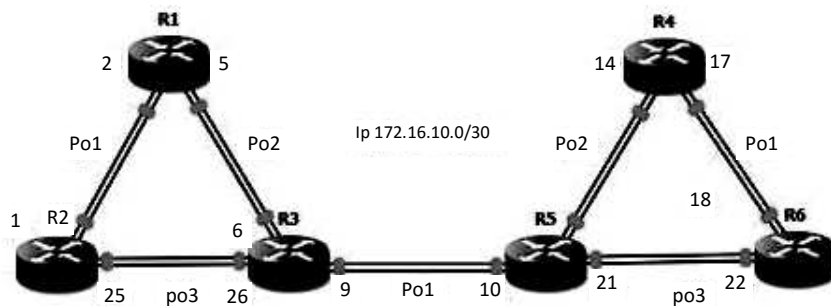
Subnet 255.255.255.252

Network $2^6 = 64$

Host $2^2 - 2 = 2$

Network	IP Address	Broadcast Address
172.16.10.0	172.16.10.1 - 172.16.10.2	172.16.10.3
172.16.10.4	172.16.10.5 - 172.16.10.6	172.16.10.7
172.16.10.8	172.16.10.9 - 172.16.10.10	172.16.10.11
172.16.10.12	172.16.10.13 - 172.16.10.14	172.16.10.15
172.16.10.16	172.16.10.17 - 172.16.10.18	172.16.10.19
172.16.10.20	172.16.10.21 - 172.16.10.22	172.16.10.23
172.16.10.24	172.16.10.25 - 172.16.10.26	172.16.10.27
172.16.10.28	172.16.10.29 - 172.16.10.30	172.16.10.31
172.16.10.32	172.16.10.33 - 172.16.10.34	172.16.10.35
172.16.10.36	172.16.10.37 - 172.16.10.38	172.16.10.39
172.16.10.40	172.16.10.41 - 172.16.10.42	172.16.10.43

2. Membuat topologi *router* pada GNS3 versi 0.8.6 di *core laptop*



po1terdiri dari g0/0 dan g1/0 po2terdiri dari g3/0 dan g5/0 po3terdiri dari g4/0 dan g2/0

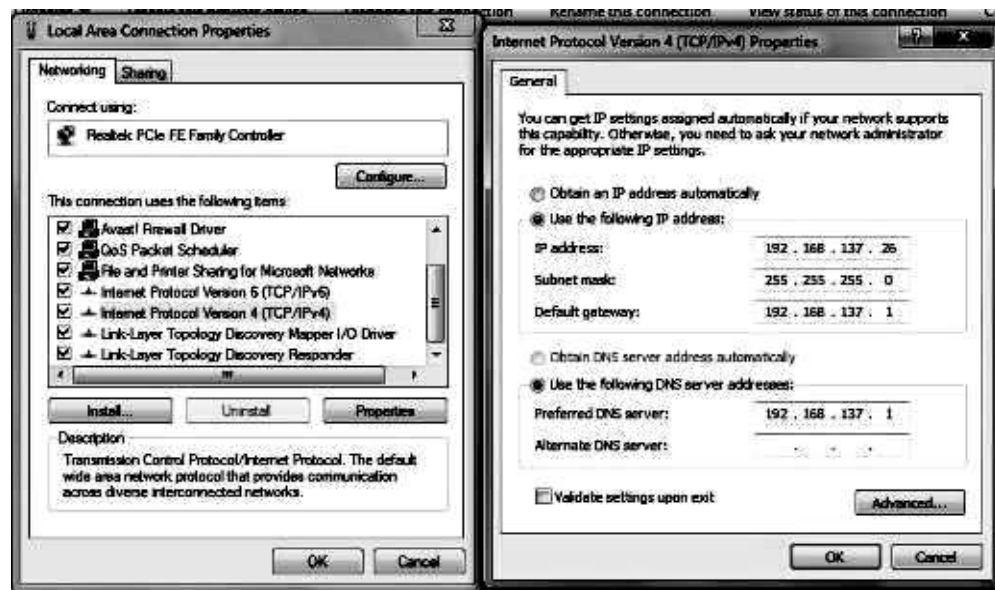
Gambar 4. Topologi *core router*

3. Konfigurasi masing-masing *router* pada *core router*, sebagai contoh pada R1 terdiri dari:

- Konfigurasi IP
- Konfigurasi *Interface*
- Konfigurasi *Routing Protokol*
- Konfigurasi *MPLS*

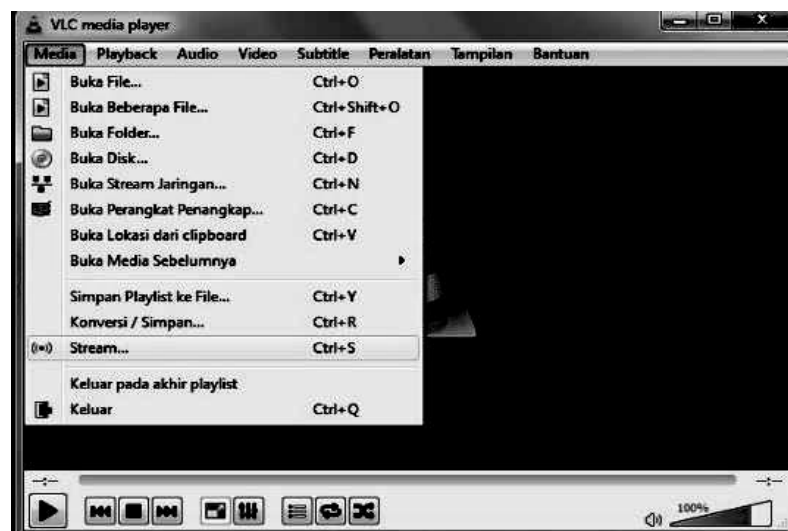
4. Konfigurasi *PC server*

Sebelum melakukan simulasi *video streaming*, *server* diberi alamat IP 192.168.137.26 sebagai identitas *server* di mana nantinya *client* dapat meminta layanan data video ke alamat IP tersebut seperti terlihat pada Gambar 5 berikut:



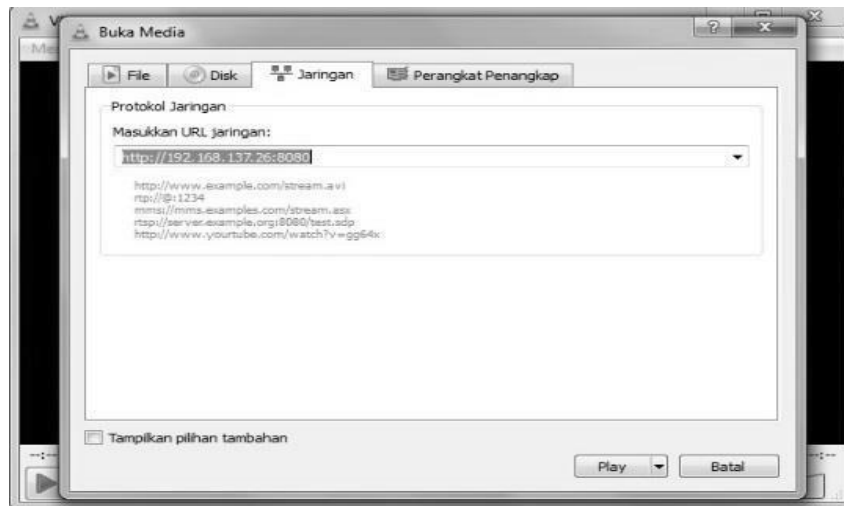
Gambar 5. Konfigurasi *PC server*

5. Memilih video yang akan di-*stream* pada laptop *server*



Gambar 6. Memilih *video streaming* di *server*

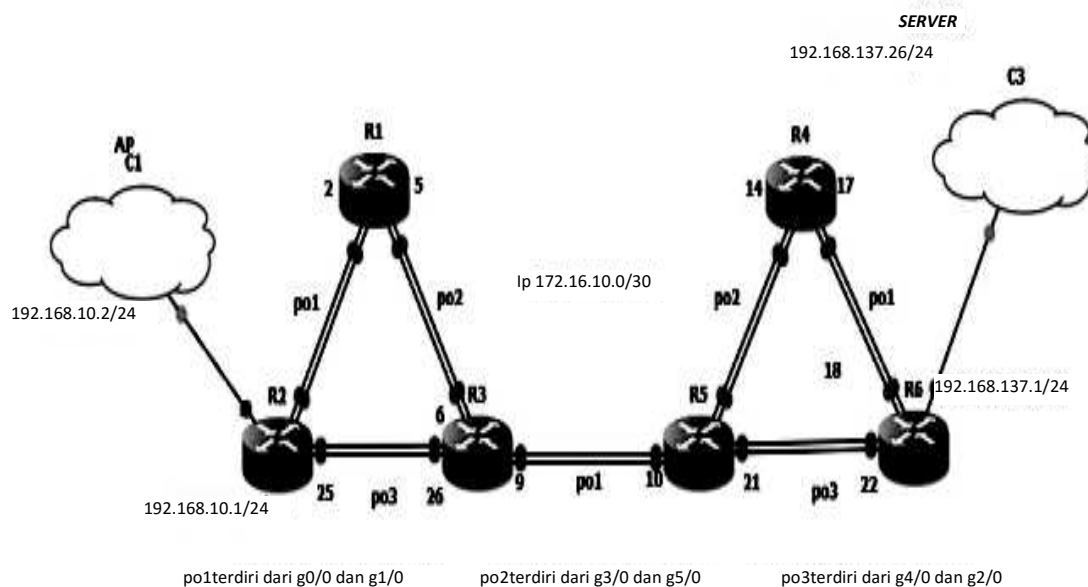
6. Membuka *streaming* jaringan dengan permintaan akses ke *server* IP 192.168.137.26



Gambar 7. Akses ke *server*

4. HASIL DAN PEMBAHASAN

Topologi yang dipakai dalam simulasi ini adalah sebagai berikut:



Gambar 8. Topologi simulasi

Dari simulasi tersebut dilakukan pengukuran nilai kinerja untuk parameter *throughput* dan *delay* sebagaimana dijelaskan di bawah ini.

4.1. Throughput

Throughput adalah kecepatan transfer data antara *server* ke *client*, atau banyaknya data yang masuk melewati sistem transmisi per waktu tertentu *byte/s*. Pengukuran *throughput* dilakukan menggunakan *wireshark* untuk layanan data video. Pengukuran ini berdasarkan angka *real* pada saat proses jaringan sedang berjalan. Berikut ini adalah hasil pengukuran *video streaming* dari sisi server sebagai *upstream* dengan perbandingan *wireless* dan *wireline*.

4.1.1. Throughput Server Wireless

Display

Display filter:

Ignored packets:

none

0 (0,000%)

Traffic	Captured	Displayed	Displayed %	Marked	Marked %
Packets	48954	48954	100.000%	0	0,000%
Between first and last packet 345,747 sec					
Avg. packets/sec	141,589				
Avg. packet size	1016,517 bytes				
Bytes	49762584	49762584	100.000%	0	0.000%
Avg. bytes/sec	143927,573				
Avg. MBit/sec	1,151				

Help

Gambar 9. *Throughput server wireless*

4.1.2. Throughput Server Wireline

Display					
Display filter:					none
Ignored packets:					0 (0,000%)
Traffic	Captured	Displayed	Displayed %	Marked	Marked %
Packets	31421	31421	100,000 %	0	0,000 %
Between first and last packet 220,268 sec					
Avg. packets/sec	142,649				
Avg. packet size	1019,002 bytes				
Bytes	32018076	32018076	100,000 %	0	0,000 %
Avg. bytes/sec	145359,862				
Avg. MBit/sec	1,163				
Help					

Gambar 10. *Throughput server wireline*

4.1.3. Throughput Client Wireless

Display					
Display filter:					none
Ignored packets:					0 (0,000%)
Traffic	Captured	Displayed	Displayed %	Marked	Marked %
Packets	51729	51729	100.000%	0	0,000%
Between first and last packet 369,136 sec					
Avg. packets/sec	140,135				
Avg. packet size	1022 bytes				
Bytes	52876777	52876777	100.000%	0	0,000%
Avg. bytes/sec	143244,639				
Avg. MBit/sec	1,146				

Gambar 11. Throughput client wireless

4.1.4. Throughput Client Wireline

Display					
Display filter:					none
Ignored packets:					0 (0,000%)
Traffic	Captured	Displayed	Displayed %	Marked	Marked %
Packets	14213	14213	100.000%	0	0,000%
Between first and last packet 99,647 sec					
Avg. packets/sec	142,633				
Avg. packet size	1018 bytes				
Bytes	14467921	14467921	100.000%	0	0,000%
Avg. bytes/sec	145191,379				
Avg. MBit/sec	1,162				

Gambar 12. Throughput client wireline

Dari data di atas disimpulkan bahwa penggunaan sistem transmisi *wireline* lebih baik dibandingkan *wireless*. Kesimpulan itu ditunjukkan dari nilai *throughput* yang dihasilkan *wireline* lebih tinggi dari pada *wireless*, baik itu pada sisi *server* ataupun sisi *client*. Nilai *throughput* yang dihasilkan pada simulasi video ini rata-rata berkisar 1 Mbit/sec. Di sini digunakan interface *gigabit ethernet* dan *fast ethernet* dengan *wireless access point* tipe G dengan *bandwidth* 54 Mbps dan PC RAM 2Gb

untuk *core router*. File video yang di-stream berkapasitas 900Mb dengan durasi 2 jam sehingga hal ini mempengaruhi kecepatan transfer.

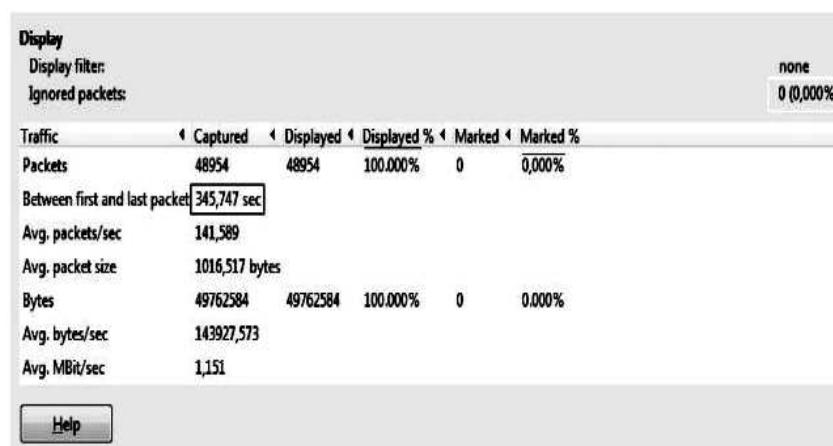
4.2. Delay

Delay adalah waktu tunda suatu paket yang diakibatkan oleh proses transmisi dari satu kanal ke kanal lain dengan tujuan pengiriman paket data. Berikut ini adalah hasil dari pengukuran *wireshark* untuk *delay*. Dari *delay* akan terlihat kualitas simulasi video *streaming* dan kualitas jaringan itu sendiri melalui menghitung *delay* rata-rata dengan membagi total *delay* / total paket yang diterima. Di sini juga dibandingkan kinerja sistem transmisi *wireless* dan *wireline*. Adapun standar referensi yang digunakan adalah standar THIPON dengan nilai sebagai berikut:

Tabel 1. Standar *delay* menurut Thipon

Kategory Delay	Besar Delay
Sangat Bagus	<150 ms
Bagus	150 ms s/d 300 ms
Sedang	300 ms s/d 450 ms
Jelek	>450 ms

4.2.1. Hasil pengukuran dari sisi *server* sebagai *upstream* dengan perbandingan *wireless* dan *wireline* seperti pengukuran *delay server wireless* pada Gambar 13 berikut.



Display					
Display filter:					none
Ignored packets:					0 (0,000%)
Traffic	Captured	Displayed	Displayed %	Marked	Marked %
Packets	48954	48954	100.000%	0	0,000%
Between first and last packet	345,747 sec				
Avg. packets/sec	141,589				
Avg. packet size	1016,517 bytes				
Bytes	49762584	49762584	100.000%	0	0,000%
Avg. bytes/sec	143927,573				
Avg. MBit/sec	1,151				
Help					

Gambar 13. Pengukuran *delay server wireless*

Perhitungan rata-rata *delay server wireless*

$$= 345,747 / 48954$$

$$= 7.06 \text{ ms}$$

Dengan cara pengukuran yang sama diperoleh hasil pengukuran *delay* pada *server* dan *client* untuk media *wireless* maupun *wireline* seperti diperlihatkan pada Tabel 2.

Tabel 2. *Delay* pada *server* dan *client*

Pengukuran	Rata-rata Delay
Server Wireless	7.06 ms
Server Wireline	7.01 ms
Client Wireless	7.1 ms
Client Wireline	7.01 s

5. KESIMPULAN

1. Hasil simulasi video *streaming* sudah memenuhi syarat sesuai dengan standar TIPHON.
2. *Throughput* yang dihasilkan pada *wireline* lebih besar dari pada *wireless*
 - *Throughput wireless client* 1.146 Mbit
 - *Throughput wireline client* 1.162 Mbit
 - *Throughput wireless server* 1.151 Mbit
 - *Throughput wireline server* 1.163 Mbit
3. Standar *delay* hasil simulasi sudah sesuai TIPHON, baik pada *wireline* maupun *wireless*, yaitu memiliki rata-rata *delay* 7 ms atau kategori sangat bagus.

DAFTAR PUSTAKA

- [1] Santekno. "RIP (Routing Information Protocol)". Internet: <http://santekno.blogspot.co.id/2013/01/pengertian-rip-routing-information.html>, 11 Januari 2013 [9 April 2015, 20:24].
- [2] Albertus Toni Setiawan. "RIP (Routing Information Protocol)". Internet: <http://redugm.blogspot.com/2011/04/rip-routing-information-protokol.html>, 20



-
- April 2011 [9 April 2015, 20:32].
- [3] Adhyatma. “Protokol Routing”. Internet: <http://adhycupu.blogspot.co.id/>, 23 Juni 2011 [9 April 2015, 20:37].
- [4] Didha Dewananta, “GNS3 Simulator Jaringan”. Internet: <http://ilmukomputer.org/2013/01/29/gns3/>, 29 Januari 2013 [9 April 2015 20:53].
- [5] Saputro Joko, *Praktikum CCNA di Komputer Sendiri menggunakan GNS3*, Jakayakarsa , Mediakita, 2010.
- [6] Astandro Koesriputranto, “Apa itu Wireshark”. Internet: <http://akfive.blogspot.co.id/2013/04/apa-itu-wireshark-apa-kegunaannya-dan.html>, 18 April 2013 [9 April 2015 20:58].